New Entity Identification Approaches in Entity Linking Systems

Priya Radhakrishnan
IIIT Hyderabad
priya.r@research.iiit.ac.in

Manish Gupta
Microsoft, India
gmanish@microsoft.com

Vasudeva Varma
IIIT Hyderabad
vv@iiit.ac.in

Abstract

In our day-to-day life, we encounter new and interesting entities (e.g., a person's name or a geographic location) while reading text. New Entity Identification (NEI) is the process of automatically identifying an entity present in text, but not present in a Knowledge Base (KB). Understanding NEI approaches is critical in the automatic construction and maintenance of KBs.

In this study we review the literature on NEI approaches of Entity Linking (EL) systems. We examine recent findings, best-result algorithms, and state-of-the-art systems pertinent to NEI research, while assessing the reproducibility of the results. We identify two prominent clusters of NEI approaches. Then, we re-implement approaches from both the clusters, and make several observations from our experiments. Our findings answer the following questions: How EL features impact NEI; will the use of a dedicated classifier for new entities improve NEI performance; and finally, how the standard EL systems can be improved to achieve better NEI performance.

1 Introduction

While reading natural language text, humans often encounter unfamiliar entities, such as people, organizations and geographic locations. Readers typically obtain detailed information about these new entities using web sites, such as Wikipedia [1]. This led to the development of systems that automatically identify entity mentions in a text document and link it to (or "ground it in") an entity referent in Wikipedia. The process of linking entity mentions in a text to the corresponding entries in a Knowledge Base (KB) such as Wikipedia is called Entity linking (EL).

Definition: Entity Linking consists of three sub-tasks (i) Mention detection - detecting the *linkable* phrases i.e. phrase that qualifies as a link to an entity in the KB, called *mentions*. (ii) Disambiguation - identifying relevant entities from a KB. (iii) Linking - choosing the most suitable entity to link the mention. If the KB does not have an entry to link the mention, then the mention is generally referred to as *NIL entity*. We refer to this task of identifying NIL entities as **New Entity Identification (NEI)**.

Motivation: NIL entities or sparsity in KBs has been recognized as an important issue in recent research [39]. Detecting NIL entities is important also to avoid creating spurious links. In this paper *we study the New Entity Identification approaches of various Entity Linking systems and propose ways to improve standard EL systems to gain better NEI performance, based on learnings from our re-implementations.* To the best of our knowledge this is the first study focusing on the NEI approaches of EL systems.

Overview of our approach: We approach the study of NEI by first analyzing existing NEI approaches in literature and then reproducing state-of-the-art approach and results. On reviewing the literature on NEI approaches (in Section 2), we found two types of NEI approaches. We call them *Thresholding* and *NIL Classification* in this paper. After analyzing 31 NEI approaches, we have selected two representative algorithms from both NEI approach types, namely TAGME [7] and the system proposed by Ploch [31] for re-implementation. We chose these systems since they are seminal work, reported the best results and to gain insights in reproducing state-of-the-art approach and results. We discuss more about our selection in Section 3 which also explains experimental setup, datasets and evaluation of re-implementations (in Section 4). We draw from the re-implementations, the pros and cons of the two NEI approach types (in Section 5) and derive conclusions (in Section 6).

Learnings from re-implementations: From the analysis of our re-implementations, we find that the NEI performance of standard EL systems can be improved by the use of dedicated NIL entity classifiers which use word features. On EL systems using thresholding, mentions predicted as NIL with high linking confidence ρ, are good NIL entity candidates. Choosing a higher NIL threshold (τ) value for the classifier helps in filtering out noise from NIL entities.

[1] Wikipedia adds 800 new articles per day, 75% [29] of this is named entities (source `https://en.wikipedia.org/wiki/Wikipedia:Statistics`)

Contributions:

- We review and analyze the main approaches of NEI in EL systems and the features used.
- We re-implement representative algorithms from both NEI approach types to get insights from reproducing state-of-the-art results and identifying improvements.
- We have made all data-sets and software used in this paper publicly available [2].

Terminology: In the literature on NEI, different terms are used by different authors to refer to entity mentions that do not have a referent entry in KB. While Bunescu and Pasca [1] refer to an entity that is not covered in Wikipedia, as *out-of-Wikipedia* entity, Hoffart et al. [13] call them *Emerging Entities* (EEs) or *out-of-knowledge-base* (OOKB) entities, Lin et al. [23] refer to them as *unlinkable-noun-phrase* and Kulkarni et al. [18] label it "NA" denoting *no attachment*. The popular TAC workshops refer to them as *NIL entities*, which we will use in this paper.

2 Review of NEI Literature

Early research on NEI [13, 32] aimed at enhancing or maintaining automatically constructed KBs. Research focus on NEI (and EL in general) was enhanced by the Text Analysis Conference (TAC) workshop's Knowledge Base Population (KBP) track [14, 15, 25] which was aimed at discovery of information for inclusion in an existing KB. TAC workshop provided standard dataset and evaluation measures to compare EL and hence NEI tasks. In this section we look at the main NEI approaches in EL literature including the TAC KBP workshop. We also look at major features used for NEI.

NEI approach types: In an extensive survey of EL systems, Wang et al. [34] identify the main approaches to NEI. A simple heuristic approach is one where if the candidate entity set $\mathbf{e}$ generated for the mention m is empty, the EL system links m to NIL. This approach is implemented in [37, 38]. The non-heuristic approaches are of two types.

Thresholding or NIL threshold method for NEI: Generally, EL systems do mention detection, disambiguation and linking sequentially. The EL system relies on confidence of the disambiguation sub-task for linking a mention to an entity. It learns a threshold value for this confidence called NIL threshold τ, from the training data. Based on τ value, EL system could drop a mention (in a low-confidence situation [4]) by declaring it unlinkable or map it to a global NIL [6]. Many systems [1, 8, 18, 20, 22, 30, 35, 36] use thresholding method to identify NIL entities. Systems like TAGME [7] and Jin et al. [17] also use a learned threshold to link the mention to NIL.

Supervised machine learning techniques: Many EL systems use supervised machine learning techniques to identify the NIL entity. Here the NEI task is, given a mention m and set t of candidate entities from the KB, identifying that m cannot be linked to any entity in t. Based on machine learning technique used, Roth et al. [33] re-group these approaches into two. First one involves classification and local processing where mention m is assigned to NIL when m cannot be linked to any entity in t. The systems [11, 31, 40, 41, 43] use a binary classifier for this. Second one involves clustering [3] and global processing. All mentions m that represent the same entity are clustered. If the cluster cannot be linked to any entity in the KB, then it is mapped to NIL [28].

NEI Features: We now look at prominent features used for NEI.

Word features: Identifying words and phrases that identify the new entity is a prominent method to do NEI. In their work on 'No-Noun Phrase' identification, Lin et al. [23] devise a supervised classifier trained on temporal features of words to predict if a noun phrase contains an entity mention. They also predict the entity's fine-grained type. Graus et al. [9] present an unsupervised method for generating pseudo-ground truth for training a named entity recognizer to specifically identify new entities that will be added to a KB. The approach of Ratinov et al. [32] is to initially rank the candidate entities by local features including keyphrases and later link to an entity or NIL. In their system Hoffart et al. [13] extract keyphrases from non-KB sources like new articles and explore the link-ability of the mention to the NIL entity with high precision.

Presence in macroKB: Another way of establishing the absence of an entity in a KB is by verifying its presence in a larger (or macro) KB. This technique is used in SIEL [38], Zhang [42] and MS_MLI [5] systems. As the TAC KB is based on the 2008 Wikipedia snapshot, a KB based on a later Wikipedia version functions as the macroKB in this case.

NEI Approaches in TAC: In this section we look at NEI approaches of EL systems that either participated in TAC or used the TAC KB dataset in their experiments. The TAC KBP track is being conducted since 2009. We analyze the NEI approaches over the years in the context of approach types and features discussed so

[2] https://github.com/priyaradhakrishnan0/NEI

[3] Though [33] calls this Entity Linking we will refer to it as Clustering in order to differentiate it from the EL system.

far.

Among TAC 2009 systems that report best NEI performance, thresholding is used by Dredze et al. [6]. They consider absence (i.e., the NIL candidate) as another entry to disambiguate and learn the τ. They use word similarity features as NEI features. Presence in macroKB feature is used by the top scoring system (SIEL) [38], which computes similarity of query to KB entities and Wikipedia. If the query has no (or very small) similarity to KB entities and has high similarity to a single Wikipedia page, it infers that the likely link for the query is not present in KB, thus it is NEI. Li et al.'s system [21] uses supervised machine learning for NIL entity detection. They first rank possible candidate entities and select the top-ranked option. Then they use a separate binary classifier to decide whether this top prediction is NIL.

Zhang et al. [42] use the presence in macroKB feature for NEI. They use Wikipedia data curated from the 2009 snapshot of Wikipedia. If the linking entity e was found in this Wikipedia data and not in TAC KB, it was declared a NIL mention. While they achieved a micro-averaged-accuracy of 0.83 for NIL entity, Ploch [31] reported the highest micro-averaged accuracy for NIL entity using TAC KB and TAC 2010 dataset at **0.96**. They approach disambiguation and NIL detection as supervised classification tasks and use two binary SVM classifiers. The first classifier decides for each candidate, if it corresponds to the target entity and second classifier detects NIL entities.

Use of supervised ML technique approach was needed from TAC KPB 2011, as participating systems had to cluster the NIL entity mentions i.e. when multiple mentions in a given document correspond to the same entity which is outside the KB, cluster the relevant mentions as representing a single NIL entity. Hierarchical clustering approach with multiple steps was adopted by the top team (LCC [28]). It used a three-step process of grouping likely matches, clustering within those groups, and merging the final clusters. Evaluation of NIL clustering was done using B-Cubed F-measure [14] and LCC system achieved a score of 0.86 on NIL entities. In TAC 2012, evaluation of NIL clustering was done using modified B-Cubed (B-Cubed+) metric [24] and top team (B_CUNY) achieved a score of 0.789 on NIL entities. The B_CUNY [37] system used collaborative clustering to achieve NIL clustering. The B_CUNY [37] system explored more than 40 clustering algorithms and found that advanced graph-based clustering algorithms did not significantly out-perform single baseline clustering algorithm on the overall queries generally. However, in TAC 2014 the top team, LCC [27] proved that graph partition based algorithm achieved gains for NIL clustering.

Using the presence in macroKB feature, TAC 2013 top team (MS_MLI) constructed a new KB by processing 2013 Wikipedia snapshot. Mentions disambiguated to entities in the new KB and having no corresponding entities in the TAC KB were labeled as NIL. In 2015, the task was extended from monolingual to trilingual, where EL systems were required to cluster mentions into NIL entities across languages [16]. New datasets (trilingual) and new KB (based on Freebase) were used. In this paper we have considered only mono-lingual (English) EL system's NEI performance based on TAC KB 2009.

3 Re-implementating Best Result and State-of-the-art Approaches

We analyzed 31 NEI approaches in Section 2. Most of the approaches we discussed are evaluated on TAC. However all approaches (e.g., TAGME) were not evaluated on the TAC dataset. To have comparable results, we re-implemented the NEI-approaches. We chose NEI approaches based on: (i) Best results - We pick the approach that reports best result on common and/or comparable test conditions, (ii) State-of-the-art - We pick the work that is novel and widely cited, (iii) Ease of reproduction - Focus on issues raised only in reproduction.

We first analyze the approaches and group them (non-heuristic approaches) into two groups namely thresholding and supervised machine learning techniques. From the thresholding approaches group we picked TAGME [7], a seminal work. From the supervised machine learning techniques group, we picked the system proposed by Ploch [31] which reported best result on the TAC 2010 dataset. TAGME source code was not available to us initially [12], prompting this re-implementation. Later it was made publicly available. Use of TAGME API did not suit our cause as API gives linking entities, whereas we are interested in the NIL entity task. We also could not obtain Ploch's source code.

We tried to pick approaches with different set of features. However some amount of feature overlap was unavoidable as the features on KB similarity was used by almost all approaches. In this section, we present a detailed analysis of the two re-implementations. These approaches have not been compared against each other before. An overview of the comparison is presented in Table 1.

TAGME System: TAGME was proposed by Ferragina et al. [7] for linking entities in short documents. Wikipedia inlinks are explored for detecting and linking the entities. Fig. 1 gives an overview of the TAGME

Approach		**Thresholding**	**Supervised ML methods**
Representative Algorithm		TAGME [7]	Ploch [31]
Mention Detection		Inverted Index Lookup	Dictionary lookup
KB features	Title	✓	✓
	Redirect		✓
	Disambiguation		✓
	Inlink	✓	✓
	Outlink	✓	✓
Disambiguation Features		Relatedness score, Prior Probability	Context Similarity, BOW, Popularity
Linking		Coherence score	NIL classification

Table 1: NEI Approaches: We analyze and group the approaches into two groups, viz. thresholding and supervised machine learning. Tabulated above are representative approaches from the groups, compared against each other on five aspects. Please see Section 3 for more details.

system re-implementation.

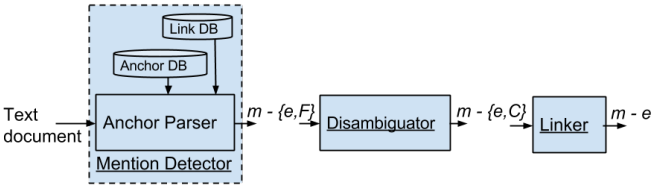

Figure 1: TAGME System : Representative of the Thresholding NEI approach. Mentions predicted as NIL with high linking confidence ρ, were found to be good NIL entity candidates.

Mention Detector: Mentions are detected using link probability $lp(m)$, which is the probability that mention m, is used as anchor in Wikipedia. Continuous word sequences of up to six words, are checked for their presence as an anchor in Wikipedia. If the $lp(m)$ of the string is greater than a predefined threshold, then it is taken as a detected mention and all pages referred by it are taken as candidate entities **e**. However this involves a large number of look-ups on Wikipedia anchor index. In order to reduce the number of look-ups, we used stopword filtering [4]. We use the standard JMLR stopword list [5].

Disambiguator: The disambiguation is done based on the probability of the mention linking to a particular entity and the Wikipedia-based semantic relatedness measure (δ) [26]. The disambiguator computes a score for each candidate e, for each mention m, based on agreement between the entities e of a mention with entities **e** of other mentions detected using δ. Linearly combining this score with the prior probability $Pr(e|m)$, of e, we get the disambiguation score of an entity.

Linking: The disambiguation phase produces one candidate entity e_m, per mention m of the input text, T. The average semantic relatedness between the candidate entity e_m and the candidates e_n assigned to all other anchors n in T is measured as Coherence. Linking combines coherence with link probability $lp(m)$ of the mention m to arrive at linking confidence ρ. The mentions with ρ value less than the threshold are NIL entities. Threshold value of ρ for NIL entities τ, is learned with an SVM classifier using the training dataset.

Ploch System: Ploch [31] approaches linking as a supervised binary classification problem using two binary SVM classifiers, one for entities present in KB and other for NIL entities. Fig 2 depicts our re-implementation of this system.

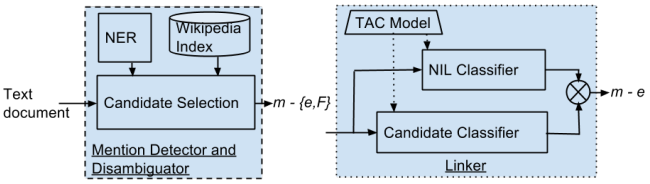

Figure 2: Ploch System: Representative of Supervised ML Method NEI Approach. Choosing a higher NIL threshold (τ) value for the classifier helps filtering noise from NIL entities.

Mention Detection and Disambiguation: Mention detection is done by dictionary look-up. A KB is created by processing Wikipedia, mapping Wikipedia article name as entity name and mapping its surface forms, categories and context words. Mention detection uses this KB to generate candidate entities **e**. The disambiguation features include Entity Context, Link Context and Standard features (including Bag-Of-Words and *tf.idf*).

[4] If the mention identified contains only stopwords, we ignore that mention

[5] `http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop`

Linker: The first classifier (Candidate classifier) decides for each candidate e, if it corresponds to the target entity. Each candidate is represented as a vector **F** of features. For training the classifier, we label at most one e from **e** as a positive example and all others as negative. A second classifier (NIL classifier) is trained to detect NIL queries. Positive samples are mentions that link to NIL and mentions that have similarity values of all candidates **e** as very low. Other features implemented were the maximum, mean and minimum, the difference between maximum and mean, and the difference between maximum and minimum, for all atomic features, using the feature vectors of all candidates **e**. Both classifiers [6] use a Radial Basis Function (RBF) kernel, with parameter settings of C = 32 and $\gamma = 8$.

Experiments and Evaluation of Re-implementations

Dataset: We use TAC dataset for evaluation. TAC evaluation dataset has queries and gold standard. The queries consist of a mention string and a source document containing it. The gold standard is a reference to a TAC KB node or NIL if there is no corresponding node in the KB. TAGME system uses training data to estimate the NIL threshold, τ. Ploch system used the TAC 2009 data set for training. So we use TAC 2009 dataset as our development dataset. We use the TAC 2010 test data as our final held-out test set. Both the 2009 and 2010 test data set has approximately 55% NIL entities, which makes our final test data not very different in the constitution of NIL entities from the training data. TAC 2009 and 2010 datasets have highest number of NIL queries compared to other TAC datasets (2011, 2012, 2013) and the AIDA_EE GigaWord dataset [13].

Evaluation Measures: Hachey et al. [10] define NIL Precision ($\mathrm{P}_\emptyset$) and NIL Recall ($\mathrm{R}_\emptyset$) as the evaluation measures for measuring NEI performance. Micro averaged Accuracy (A_{micro}), which is percentage of correctly linked queries, is the official TAC measure for evaluation of EL systems. TAC reports NIL accuracy ($A_\emptyset$) which is $\mathrm{R}_\emptyset$ calculated with the system generated entity set having a single entity which is NIL entity. These are calculated as follows:

$$\mathrm{A}_{micro} = \frac{|\{S_{i,0}|S_{i,0} = G\}|}{Q} \tag{3.1}$$

$$\mathrm{P}_\emptyset = \frac{|\{S_i|S_i = \emptyset \wedge G_i = NIL\}|}{|\{S_i|S_i = \emptyset\}|} \tag{3.2}$$

$$\mathrm{F} = \frac{2P_\emptyset R_\emptyset}{P_\emptyset + R_\emptyset} \tag{3.3}$$

$$\mathrm{R}_\emptyset = \frac{|\{S_i|S_i = \emptyset \wedge G_i = NIL\}|}{|\{G_i|G_i = NIL\}|} \tag{3.4}$$

where Q is the number of queries in the dataset, G is the gold standard annotations for the dataset ($|G| = Q$), G_i is the gold standard for query i (KB ID or NIL), S is the system generated entity sets ($|S| = Q$), S_i is the system generated entity set for i^{th} query, $S_{i,j}$ is the system generated entity at j^{th} rank of i^{th} query, $P_\emptyset$ is the percentage of system generated NIL entity sets (sets that are either empty or singleton containing NIL) that are correct (which correspond to NIL queries), and $R_\emptyset$ is the percentage of NIL queries for which the system generated NIL entity sets.

Cornolti et al. [3] have defined many evaluation measures for wikifier (*A2W* systems). The precision and recall measures specified there are in line with the $\mathrm{P}_\emptyset$ and $\mathrm{R}_\emptyset$ evaluation measures respectively. Further Hoffart et al. [13] define *EE Precision* and *EE Recall* which coincides with $\mathrm{P}_\emptyset$ and $\mathrm{R}_\emptyset$ respectively.

Evaluation with KBs: NEI can be evaluated using two KBs with one KB being a subset of the other. Entity Linking is performed with a smaller KB and the NIL entities identified are evaluated by their presence in the larger (superset) KB. This strategy is used in many EL systems for NEI [5, 38, 42] and NEI evaluation [4, 9, 42]. We use the evaluation setup specified by Graus et al. [9]. Given the TAC KB, we randomly sample entities to yield a smaller KB referred as KB_s. Now KB_s simulates the available knowledge at the present point in time, whilst KB represents the future state. By measuring how many entities we are able to detect in our corpus that feature in KB, but not in KB_s, we can measure NEI. KB_s is created by taking random samples of 20% to 100% the size of KB (measured in entities), in steps of 20% (as a sliding window). We repeat each sampling step five times to avoid bias.

4 Reproducing NEI Approaches

Reproducibility: Table 2 shows A_{micro}, $\mathrm{P}_\emptyset$ and $\mathrm{R}_\emptyset$ for the two NEI approaches on the test set. Threshold value for NIL entity (τ) for TAGME system, learned with an SVM regression classifier trained on the NIL queries of development dataset, was 0.22. This is in line with the TAGME description [7]. We evaluated Ploch system with both linear and RBF kernels. Ploch reports A_{micro} of 0.62 for KB queries (queries for entities present in KB). While we got the same result, we found that moving from linear kernel to RBF kernel, A_{micro} dropped from 0.62 to 0.34 for KB queries. On NIL queries, Ploch reports A_{micro} of 0.88 with RBF kernel. In our implementation we found this to be 0.52

[6] We used the libsvm implementation `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

NEIApproach	A_{micro}	$P_{\emptyset}$	$R_{\emptyset}$	F-measure
$Ploch_{KBlin}$	0.62	0.63	0.89	0.74
$Ploch_{NILlin}$	0.44	0.70	0.81	0.75
$Ploch_{KBrbf}$	0.34	0.63	0.49	0.55
$Ploch_{NILrbf}$	0.52	0.60	0.94	0.73
$TAGME_{\rho}$	0.47	0.62	0.80	0.70
$TAGME_{\tau}$	0.45	0.62	0.73	0.67

Table 2: NEI Performance: Ploch system performance on NIL entities is better than that in-KB entities and also that of TAGME system. Use of dedicated NIL entity classifiers which uses NEI word features improves NEI performance of standard EL systems.

with RBF kernel. We tried with the relaxed evaluation conditions as suggested in [12], which took us near to the 0.88 result. But deviating from the evaluation measures defined in Section 3 will leave us with results that are not comparable. So we use 0.52 as A_{micro}. On moving from RBF kernel to linear kernel, A_{micro} dropped from 0.52 to 0.44. This observation is reported in the paper [31] too. Thus both the approaches are reproducible, though all results could not be reproduced.

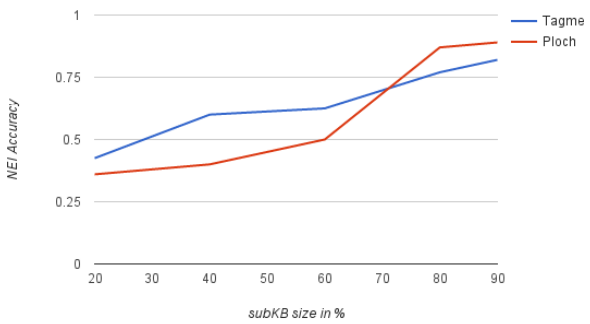

Figure 3: NEI Accuracy by KB-subKB evaluation method: the Ploch system does slightly better than the TAGME system in NEI performance by this evaluation.

Performance: As this is the first reporting of A_{micro}, $P_{\emptyset}$ and $R_{\emptyset}$ of TAGME system on this dataset, we do not have a benchmark. However we compare it to that of Ploch system performance. As TAGME system is designed to predict in-KB entities, we compare the in-KB results of Ploch system namely $Ploch_{KBlin}$ and $Ploch_{KBrbf}$ with TAGME results $TAGME\rho$. Here we see that $P_{\emptyset}$ remains in the range of 0.63 ± 0.1. $R_{\emptyset}$ remains higher in Ploch system compared to TAGME. With linear kernel Ploch system gave A_{micro} of 0.62 whereas TAGME system's A_{micro} was 0.46 ± 0.1. Comparing the performance on NIL entities the F-measure (Eq. 3) of Ploch system fares better than that of TAGME system. Further the NIL-results are better than in-KB results in Ploch system, all proving that NEI features and separate NIL classifier help achieve better NEI.

Alternate evaluation: On evaluating with subset and master-set KBs, we report the accuracy of NIL predictions as $Accuracy_{\emptyset}$. $Accuracy_{\emptyset}$ is calculated by taking the set of correct predictions (true positives), and linking each mention to the referent entity in the super-set KB. This gives the fraction of newly discovered entities. Figure 3 shows the average $Accuracy_{\emptyset}$ of TAGME and Ploch system, across the five samples on the y-axis with varying subKB sizes on the x-axis. Here we see that the $Accuracy_{\emptyset}$ varies from 36% to 89% across 20% to 90% of KB size. Similar gradual improvement was observed in precision of the systems with increasing KB_s size. Recall of the systems increased with increasing sub-KB size, which could be attributed to a broader coverage in KB. This result is in line with that of Graus et al. [9]. We also observe that Ploch system does slightly better than TAGME system in NEI performance by this evaluation.

5 Lessons Learned from Re-implementations

In this section we look at possible improvements to the NEI approaches of EL systems based on the learnings from our analysis and re-implementations.

NEI approach types: Systems using supervised machine learning method for NEI were found to perform better than systems using thresholding and systems using heuristic methods. Heuristic approaches like empty candidate sets were found to perform poorly in high recall systems. Thresholding has been used for NEI starting from NEI approach of Bunescu and Pasca [1]. Hoffart et al. [13] note three shortcomings with thresholding systems. (i) Its empirical quality is not that good. (ii) Fixing one global value for τ may be difficult and may affect NEI decisions on local mentions. (iii) Fixing different τ values for different kinds of contents may need frequent re-tuning on appropriate training data. Basically, thresholding systems make a trade-off between precision of linking the in-KB entities and NEI performance, in choosing the τ value. For example, TAGME system assigns NIL to a mention when $\rho < \tau$. In our re-implementation, we find that setting a higher τ results in more mentions predicted as NIL entity. Higher values of ρ (almost nearing τ) was found to be a good indicator of new entity. Similar result was observed by Graus et al. [9] also, who showed that higher ρ value is an effective signal to separate noise

from entities that are worth including in a KB. Thus we can conclude that, on EL systems using thresholding which are trained for optimum performance, mentions predicted as NIL entity with high value of ρ (almost nearing τ) are good candidates for improving NEI performance.

NEI Features: Word features or lexical similarity of mention and context [19] to candidate entities, is the most popular feature used in NEI approaches. We observe that word features lead to better recall while meta-data features like prior probability, link probability and coherence lead to better precision. In our re-implementation, (Table 1, row 'KB features') we find that Ploch system uses all word features, while TAGME system uses only three of them. In 'Disambiguation features', Ploch system uses more word features compared to TAGME system which uses more meta data features. In Table 2, $R_{\emptyset}$ is higher for Ploch_{NIL} (both linear and RBF) than TAGME_{τ}. The increased recall could be due to the higher word overlap with KB. We find the system with better recall does better NEI. This result was observed by Graus et al. [9] also. Thus we can conclude that EL system's NEI performance can be improved by dedicated classifier (or clusterer) using word features.

Evaluating NEI: NEI evaluation is a challenging task. Hoffart et al. [13] create a labeled dataset for the task with manual cleaning. Graus et al. [9] also report manual evaluation while Hachey et al. [2] use crowd sourcing. In this regard, NEI evaluation with two KBs, one KB being a subset of the other, is promising [4, 42] and is especially well suited for the unsupervised NEI approaches [9].

Reproducing published results : We had to adapt the disambiguation function of TAGME and define the relatedness between two pages p_a and p_b as shown in Eq. 5.5.

$$rel(p_a, p_b) = 1 - \delta(p_a, p_b) \tag{5.5}$$

This was needed as, when p_a and p_b are identical pages ($\delta(p_a, p_b) = 0$) relatedness score $rel(p_a, p_b)$ becomes 1. Otherwise $rel(p_a, p_b)$ is a score between 0 and 1. Though TAGME [7] and Milne & Witten [26] systems use δ to measure relatedness between two pages, we used 1 - δ in our system for the above reason. Later in a personal communication, TAGME team confirmed that this adaptation was correct and required.

6 Conclusion and Future Directions

New Entity Identification is the process of identifying entities to be added to a Knowledge base. In this study we analyzed 31 NEI approaches and re-implemented two state-of-the-art approaches. Our attempts to reproduce published research results have helped us to get a deeper understanding of NEI task and get insights into recreating state-of-the-art results. We have presented a systematic investigation of the NEI task, by re-implementing representative systems from both the approaches in the literature. We have presented the first direct comparison of these approaches, analyzed the results and evaluated them using both evaluation measures method and KB-subKB method. From our experiments we find that NEI performance can be improved by using a dedicated classifier (or clusterer) with word features. Crowd sourcing is a promising way for new entity evaluation.

Although there are many research efforts in NEI, we believe that there are still many opportunities for substantial improvements in this field, for instance in identifying new entities from unstructured documents. Use of deep learning techniques have shown improvements in EL performance. Research to use these improvements to enhance NEI performance is an interesting direction. Further the increasing demand for constructing and populating domain-specific knowledge bases (e.g., in the domains of bio-medicine, entertainment, products, finance, tourism) makes domain-specific NEI important as well. We hope that the findings from this paper will provide NEI researchers with a quick start in their efforts.

References

[1] R. Bunescu and M. Pasca. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *EACL*, pages 9–16, 2006.

[2] A. Chisholm, W. Radford, and B. Hachey. Discovering Entity Knowledge Bases on the Web. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction, AKBC@NAACL-HLT San Diego, CA, USA*, pages 7–11, 2016.

[3] M. Cornolti, P. Ferragina, and M. Ciaramita. A Framework for Benchmarking Entity-annotation Systems. In *WWW*, 2013.

[4] S. Cucerzan. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *EMNLP-CoNLL*, pages 708–716, 2007.

[5] S. Cucerzan. The MSR System for Entity Linking at TAC 2012. In *TAC*, 2013.

[6] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin. Entity Disambiguation for Knowledge Base Population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 277–285. Association for Computational Linguistics, 2010.

[7] P. Ferragina and U. Scaiella. TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities). In *Proc. of the* 19^{th} *ACM Intl. Conf. on Information and Knowledge Management (CIKM)*,

pages 1625–1628, 2010.

[8] S. Gottipati and J. Jiang. Linking Entities to a Knowledge Base with Query Expansion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 804–813. Association for Computational Linguistics, 2011.

[9] D. Graus, M. Tsagkias, L. Buitinck, and M. de Rijke. Generating Pseudo-Ground Truth for Predicting New Concepts in Social Streams. In *ECIR*, 2014.

[10] B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J. R. Curran. Evaluating Entity Linking with Wikipedia. *Artificial Intelligence*, 194:130–150, Jan. 2013.

[11] X. Han and L. Sun. A Generative Entity-mention Model for Linking Entities with Knowledge Base. In *HLT*, pages 945–954, 2011.

[12] F. Hasibi, K. Balog, and S. E. Bratsberg. On the Reproducibility of the TAGME Entity Linking System. In *Proceedings of the 38th European conference on Advances in Information Retrieval*, ECIR '16, pages 436–449. Springer, 2016.

[13] J. Hoffart, Y. Altun, and G. Weikum. Discovering Emerging Entities with Ambiguous Names. In *WWW*, pages 385–396, 2014.

[14] H. Ji, R. Grishman, and H. Dang. Overview of the TAC2011 Knowledge Base Population Track. In *TAC 2011 Proceedings Papers*, 2011.

[15] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis. Overview of the TAC 2010 Knowledge Base Population Track. In *In Third Text Analysis Conference (TAC)*, 2010.

[16] H. Ji, J. Nothman, B. Hachey, and R. Florian. Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking. In *In Text Analysis Conference (TAC) 2015 Proceedings*, 2015.

[17] Y. Jin, E. Kıcıman, K. Wang, and R. Loynd. Entity Linking at the Tail: Sparse Signals, Unknown Entities, and Phrase Models. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 453–462. ACM, 2014.

[18] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective Annotation of Wikipedia Entities in Web Text. In *KDD*, pages 457–466, 2009.

[19] N. Lazic, A. Subramanya, M. Ringgaard, and F. Pereira. Plato: A Selective Context Model for Entity Resolution. *Transactions of the Association for Computational Linguistics*, 3:503–515, 2015.

[20] J. Lehmann, S. Monahan, Nezda.L, Jung.A, and Shi.Y. LCC approaches to knowledge base population at TAC 2010. In *In Third Text Analysis Conference (TAC)*, 2010.

[21] F. Li, Z. Zhang, F. Bu, Y. Tang, X. Zhu, and M. Huang. THU QUANTA at TAC 2009 KBP and RTE Track. In *TAC 2009 Proceedings Papers*, 2009.

[22] Y. Li, C. Wang, F. Han, J. Han, D. Roth, and X. Yan. Mining evidences for named entity disambiguation. In *KDD*, pages 1070–1078, 2013.

[23] T. Lin, Mausam, and O. Etzioni. No Noun Phrase Left Behind: Detecting and Typing Unlinkable Entities. In *EMNLP-CoNLL*, 2012.

[24] J. Mayfield, J. Artiles, and H. T. Dang. Overview of the TAC 2012 Knowledge Base Population Track. In *In Text Analysis Conference (TAC)*, 2012.

[25] P. McNamee and H. Dang. Overview of the TAC 2009 Knowledge Base Population Track. In *In Text Analysis Conference (TAC*, 2009.

[26] D. Milne and I. H. Witten. An Effective, Low-Cost Measure of Semantic Relatedness Obtained from Wikipedia Links. In *Proceedings of AAAI 2008*, 2008.

[27] S. Monahan, D. Carpenter, M. Gorelkin, K. Crosby, and M. Brunson. Populating a Knowledge Base with Entities and Events. In *In Seventh Text Analysis Conference (TAC)*, 2011.

[28] S. Monahan, J. Lehmann, T. Nyberg, J. Plymale, and A. Jung. Cross-Lingual Cross-Document Coreference with Entity Linking. In *In Fourth Text Analysis Conference (TAC)*, 2011.

[29] J. Nothman, J. R. Curran, and T. Murphy. Transforming Wikipedia into Named Entity Training Data. In *Proceedings of the Australasian Language Technology Association Workshop 2008*, pages 124–132, 2008.

[30] A. Pilz and G. Paaß. From Names to Entities Using Thematic Context Distance. In *CIKM*, pages 857–866, 2011.

[31] D. Ploch. Exploring Entity Relations for Named Entity Disambiguation. In *Proceedings of the ACL 2011 Student Session*, HLT-SS '11, pages 18–23. Association for Computational Linguistics, 2011.

[32] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and Global Algorithms for Disambiguation to Wikipedia. In *HLT*, pages 1375–1384, 2011.

[33] D. Roth, H. Ji, M.-W. Chang, and T. Cassidy. Wikification and Beyond: The Challenges of Entity and Concept Grounding. In *ACL (Tutorial Abstracts)*, page 7, 2014.

[34] W. Shen, J. Wang, and J. Han. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *IEEE Transactions on Knowledge and Data Engineering*, 99, 2014.

[35] W. Shen, J. Wang, P. Luo, and M. Wang. LINDEN: Linking Named Entities with Knowledge Base via

Semantic Knowledge. In *WWW*, pages 449–458, 2012.

[36] W. Shen, J. Wang, P. Luo, and M. Wang. Linking Named Entities in Tweets with Knowledge Base via User Interest Modeling. In *KDD*, pages 68–76, 2013.

[37] S. Tamang, Z. Chen, and H. Ji. CUNY BLENDER TAC-KBP2012 Entity Linking System and Slot Filling Validation System. In *In Fifth Text Analysis Conference (TAC)*, 2012.

[38] V. Varma, P. Pingali, R. K, S. Krishna, S. Ganesh, K. Sarvabhotla, H. Garapati, H. Gopisetty, V. B. Reddy, K. Reddy, R. Bharadwaj, and P. Bysani. IIIT Hyderabad at TAC 2008. In *TAC*, 2009.

[39] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin. Knowledge Base Completion via Search-based Question Answering. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 515–526, New York, NY, USA, 2014. ACM.

[40] W. Zhang, C. Lim Tan, J. Su, B. Chen, W. Wang, Z. Toh, Y. Sim, Y. Cao, and C. Y. Lin. I2R-NUS-MSRA at TAC 2011: Entity Linking. In *In Fourth Text Analysis Conference (TAC)*, 2011.

[41] W. Zhang, Y. C. Sim, J. Su, and C. L. Tan. Entity Linking with Effective Acronym Expansion, Instance Selection and Topic Modeling. In *IJCAI*, pages 1909–1914, 2011.

[42] W. Zhang, J. Su, C. L. Tan, and W. T. Wang. Entity Linking Leveraging Automatically Generated Annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1290–1298. Association for Computational Linguistics, 2010.

[43] Z. Zheng, F. Li, M. Huang, and X. Zhu. Learning to Link Entities with Knowledge Base. In *HLT*, pages 483–491, 2010.